(2)

DARPA/ONR Quarterly Report

CONTRACT NUMBER:	N00014-91-J-1985
CONTRACTOR:	Duke University
ARPA ORDER NO:	4331714-01/4-6-88
PROGRAM NAME:	N00014
CONTRACT AMOUNT:	696,899.00
EFFECTIVE DATE OF CONTRACT:	July 1, 1991
EXPIRATION DATE OF CONTRACT:	June 30, 1994
PRINCIPAL INVESTIGATOR:	John H. Reif
TELEPHONE NUMBER:	(919) 660-6568
SHORT TITLE OF WORK:	Implementation of Parallel Algorithms
REPORTING PERIOD:	1 July 1991 — 30 Sept 1991

DESCRIPTION OF PROGRESS--

Special note: We are involved in a number of collaborations with other DARPA/ONR contractors: Mike Carr, of Software Options, Inc. (see item #2).

Investigations of several subproblems in the area of derivation of parallel programs were continued during the current quarter. These investigations include:

1. Mike Landis (A.B.D.) with Robert Wagner and John Reif:

Summary:

We have completed our work in the investigation of how to map context-free grammar recognition onto systolic arrays. We are currently in the final phases of the preparation of a technical report which will document this work. Our current research efforts are to extend our method to other algorithms. We are just beginning this investigation. We plan to submit our work for publication in a journal after doing this extension, as described below.

Details:
We have developed a new method for mapping algorithms into parallel architectures. This new method works very well for a class of dynamic programming problems, including CFG recognition.

In our work, we define an instance of an algorithm may be represented as a DAG. Mapping the algorithm onto a parallel architecture then is analogous to scheduling the nodes in the DAG, i.e. to assign to each computation in the algorithm a time and a location that is consistent with the architecture. To ease this task, we have proven that a restricted form of recurrence equation, called a quasi-uniform recurrence equation (QURE), is computationally equivalent to a class of systolic array architectures. The class of QUREs is a broad class, the main restriction being that the recurrence must exhibit a finite set of dependency vectors over all computations. Expressing an algorithm as a recurrence equation, the problem of mapping the algorithm onto a systolic array becomes the problem of translating the recurrence equation into QURE form. It is also very easy to translate between the DAG representation and the recurrence relation.

We have noted that intermediate results and elementary data are generally used again and again in many algorithms, especially dynamic programming algorithms. We have also noted that the task of moving these data and results around is the most constraining factor for most architectures. We have determined that the number of copies that an architecture can make of an operand in one time step to some degree characterizes the communication constraints of most architectures.

In our method, we take the original DAG and label its arcs and nodes to meet two constraints: The number of copies of each computation that can exist in each timestep, and the number of data that each computation can receive in each timestep. To implement these constraints, we proceed through several steps:

1) Derive a computational indexing scheme for the algorithm. The index enumerates all of the intermediate results in the algorithm such that each result C(i) is dependent on results C(j) where i>j. Each C(i) is called a computation point.

2) Build a time-availability table t (i,j) and vector T (j) where each t(i,j) is filled with the time that each computation point C(i) is available for use

in the evaluation of C(j), and T(j) is the minimum time at which C(j) can be computed.

In building this table, we also make use of further important constraint. We note that certain operands need to be used with other operands. We call this constraint the simultaneous availability constraint, since one operand must become available with another. This constraint directly affects the values in the table t(i,j).

3) We then augment this table with information about the source through which each datum may be passed.

4) We then invert this table to determine what data is passed on which computation points at each time.

Current investigation on the extension of our method mostly involves step 1. Deriving the computational indexing scheme is analagous to solving a specific graph layout problem. In order to achieve a QURE as the result of our method, the nodes in the DAG must be labeled in a multi-dimensional coordinate system such that the computation of each node only depends upon nodes that are a fixed distance away. We are currently investigating a shortest-path method to perform this indexing, which promises to work for a broad class of algorithms -- not just dynamic programming.

2. Sri Krishnan (graduate student) with John Reif:

M. Karr, S. Krishnan and J. Reif "Derivation of the Ellipsoid Algorithm" Draft, 1990.

Krishnan is working on derivations of the ellipsoid algorithm for linear programming, and has recently coauthored a paper on this subject with Mike Carr, another DARPA/ONR contractor that formally derives the Khachian Ellipsoid Algorithm.This paper examines some important aspects of linear programming. First we examine the basic nature of the problem and the ellipsoid algorithm, the first polynomial time algorithm for linear programming. In particular, we have rewritten the ellipsoid method in a manner that derives the algorithm more naturally. We have treated this natural derivation of the ellipsoid method as the main goal of the paper. Our approach is to derive the method from a more intuitive view point. The work that Krishnan has done (in association with Mike Karr, Software Options Inc. and John Reif, Duke University) relates to the derivation of algorithms for linear programming. It is our belief that a small well-chosen set of geometric and algebraic ideas is sufficient to derive the newer algorithms for linear programming.In particular, we have written a paper that derives the ellipsoid algorithm in this framework. This paper shows how to derive the ellipsoid algorithm from a simple geometric example that is then generalized using transforms. As a side effect,we have circumvented the tedious correctness proof of the algorithm too.

3. Doreen Yen (graduate student) with John Reif:

Yen's project is to work on formal derivation of parallel graph algorithms, currently working on generalizing the idea of "streaming contraction" from V. Ramachandran's 1988 paper "Efficient Parallel Triconnectivity in Logarithmic Time" as a derivation technique which can be applied to derive the optimal list ranking algorithm by Kosaraju and several other parallel connected components algorithms. A preliminary paper, "Derivation of Parallel Graph Algorithms via Stream Compaction," is now written.

This paper uses a denotational definition derivation to derive sequential and parallel connected components algorithms from mathematical specifications. Our initially derived sequential recursive

functional algorithm retains quantifiers and set notation. It is extracted from a denotational definition of a monotonic continuous functional whose least fixed point is a function consistent with the problem specification.

The initial algorithm is made more efficient by restricting the nondeterministic choices through a sequence of algebraic transformations, motivated by lemmas which direct the unfolding of definitions as in Reif and Scherlis, [Reif 84] and which show the base cases of inductive definitions and recursive functions may be substituted. This results in a sequential graph algorithm, Algorithm Cu, for computing the connected component which contains the vertex u.

An initial parallel connected components algorithm is derived from Algorithm Cu. The parallelism arises naturally by using a higher order function, map, which applies Algorithm Cu to each element of the vertex set. A more efficient parallel algorithm, fcc, is also obtained by denotational definition derivation. Its transition to an iterative algorithm is by discovery of lemmas showing the number of iterations needed to reach the fixed point and choice of efficient data structures to implement disjoint set union. An arbitrary constraint upon the data structure is imposed to obtain the parallel connected components algorithm connect, by Hirschberg, Chandra and Sarwate [Hirschberg 79]

We show how algorithm connect can be made more efficient by using a transformation introduced by Reif and Pan [Reif 86] which they call stream compaction, used by them in path algebra problems [Reif 88] and also independently introduced by Ramachandran and Vishkin in a triconnectivity problem [Ramachandran 88] for reducing the time complexity by O(log n).

A proof is given to show the O(log n) speed up specifically for the connected components example and another proof is given showing the speed up for any parallel algorithm computing a value at a current level and time as a function of previous levels and times. The time complexity of the final algorithm is O(log n).

4. Subhrajit Bhattacharya with John Reif:

Two efficient parallel sorting algorithms are by Cole, and Bilardi and Nicolau. An effort has been made to derive these algorithms starting from a simple definition of the problem. Special attention has been given to Cole's algorithm. These algorithms are derived starting from simple and inefficient algorithms.

5. Hillel Gazit with John Reif

An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph.

We present a parallel randomized algorithm for finding the connected components of an undirected graph. Our algorithm expected running time is $T = O(\log(n))$ with
$P = O((m+n)/\log(n))$ processors, where m is the number of edges and n is the number of vertices. The algorithm is *optimal* in the sense that the product, PT, is a linear function of the input size. The algorithm requires $O(m + n)$ space which is the input size, so it is *optimal* in space as well.

A Randomized Parallel Algorithm for Planar Graph Isomorphism

We present a parallel randomized algorithm for finding if two planar graphs are isomorphic. Assuming that we have a tree of separators for each planar graph, our algorithm takes O(log(n))

time with $P = O(n^{1.5} (\log n)^{0.5}$ processors with probability to fail of 1/n or less, where n is the number of vertices. The algorithms needs
2 log(m) log(n) +O(log(n)) random bits. The number of random bits can be decreased to O(log(n)) by increasing the processors number to $n^{3/2+e}$. This algorithm significantly improves the previous results of n^4 processors.

7. Various Authors--*Synthesis of Parallel Algorithms* (edited by Reif)

Reif has organized a large number of prestigious researchers in the field of parallel algorithms to participate in writing a textbook on algorithms synthesis. This text should draw together the many different principles which have been used to develop the current large collection of parallel algorithms which are theoretically interesting. *Synthesis of Parallel Algorithms* will be published by Morgan Kaufmann in Summer 1991.

At the present writing, some 27 researchers have submitted chapters for this text. Each author is refereeing one or two other chapters. Reif intends to collaborate with several of these researchers, and has invited them to visit Duke, where they will be available for discussion with other members of the Duke community, including the participants in the other projects funded by this contract. The design of the textbook is now underway.

This textbook promises to have significant impact on the development of parallel algorithms in the future. It should also serve as a central source, from which the details of the derivation process for some classes of algorithms can be extracted, and turned into a tool-set useful for developing future algorithms.

In inviting participation, Reif suggested that each chapter begin with a careful statement of the fundamental problems, and the solution and analytic techniques to be used in their solution. He suggested that these techniques be related, where possible, to known efficient sequential algorithms. In later sections of the chapter, more sophisticated parallel algorithms are to be synthesized from the simpler parallel algorithms and techniques discussed earlier. Thus, a progression from simple to more complicated (and presumably more efficient) algorithms would be created. This progression should reveal the kinds of transformations needed in synthesizing parallel algorithms.

Participating Authors and Topics

Guy Blelloch	Prefix Sums and Applications
Margaret Reid-Miller	Parallel Tree Contraction and Applications
Sara Baase	Introduction to Parallel Connectivity, List Ranking, and Euler Tour Techniques
Uzi Vishkin	Advanced Parallel Prefix-sums, List Ranking and Connectivity
Hillel Gazit	Randomized Parallel Connectivity
Vijaya Ramachandran	Parallel Open Ear Decomposition with Applications to Graph Biconnectivity and Triconnectivity
Vijay Vazirani	Parallel Graph Matching
Erich Kaltofen	Dynamic Parallel Evaluation of Computation DAGs
Jeffrey Ullman	Parallel Evaluation of Logic Queries
Philip Klein	Parallel Algorithms for Chordal Graphs

Participating Authors and Topics (continued)

Victor Pan	Parallel Solution of Sparse Linear and Path Systems
Andrew Goldberg	Parallel Algorithms for Network Flow Problems
Stephen Tate	Newton Iteration and Integer Division
Joachim von zur Gathen	Parallel Linear Algebra
Dexter Kozen and Doug Ierardi	Parallel Resultant Computation
Richard Cole	Parallel Merge Sort
Mikhail Atallah and Michael Goodrich	Deterministic Parallel Computational Geometry
Sandeep Sen and Sanguthevar Rajasekaran	Random Sampling Techniques and Parallel Algorithms Design
Philip Gibbons	Asynchronous PRAM Algorithms
Raymond Greenlaw	Polynomial Competeness and Parallel Computation
Baruch Schieber	Parallel Lowest Common Ancestor Computation
Faith Fich	The Parallel Random Access Machine

8. Duke Algorithm Derivation Seminar:

Participants--Professors Robert Wagner, Donald Loveland, Gopalan Nadathur and John Reif; visiting guest speakers in attendance were: Greg Plaxton, MIT; Uzi Vishkin, Maryland; Vijay Vazirani, Cornell; Awok Aggarwal, IBM; Pankal Agarwal, Duke/DIMACS; Jim Storer, Brandeis; Sampath Kannan, Berkeley; Weizhen Mao, Princeton; Satish Rao, Harvard; and, Andrew Yao, Princeton.

9. Researchers supported (other than PI):

Subhrajit Bhattacharya, graduate student
Srinivasan Krishnan, graduate student
Mike Landis, graduate student
Lars Nyland, graduate student, then post-doc
Sandeep Sen, graduate student, then post-doc
Doreen Yen, graduate student
Hillel Gazit, professor
Ming Kao, professor
Robert Wagner, professor

10. Degrees awarded:

Sandeep Sen received his Ph.D. from Duke and was here as a post-doc. Steve Tate and Lars Nyland received their Ph.D.s in January 1991 under Reif and are both remaining at Duke as post-docs.

11. Papers

H. Gazit, "An Optimal Randomized Parallel Algorithm for Finding Connected Components in a Graph," (extended version), accepted to SIAM J. of Computing, 1990.

H. Gazit, "An Optimal O(log n) Determininistic EREW Parallel Algorithm for Finding Connected Components in a Low Genus Graph," submitted for publication, 1990.

H. Gazit, "Finding the Diameter of a Directed Graph," submitted for publication, 1990.

H. Gazit, "Parallel Algorithms for Connectivity, Ear Decomposition and st-Numbering of Planar Graph," accepted to the Fifth International Parallel Processing Symposium, 1990.

Nyland, L., "The Design of a Prototyping Programming Language for Parallel and Sequential Algorithms," Ph.D. Thesis, Duke University, 1990.

Tate, Steve, "Arithmetic Circuit Complexity and Motion Planning," Ph.D. Thesis, Duke University, 1990.

R. Wagner and M. Landis, "Mapping Algorithms into VLSI Arrays through Computational Dependency Graph Refinement", submitted for publication, 1990.

J. Reif and S. Bhattacharya, "Derivation of Efficient Parallel Sorting Algorithm", draft, 1990.

J. Reif, D.W. Blevins, E.W. Davis, and R.A. Heaton. BLITZEN: A Highly Integrated Massively Parallel Machine. J. of Parallel and Distributed Computing, 150-160, 1990.

J. Reif and V. Pan, "On the Bit Complexity of Discrete Approximations to PDEs", *International Colloquium on Automata, Languages, and Programming*, Springer-Verlag Lecture Notes in Computer Science, Warwich, England, July 16-20, 1990.

J. Reif and J. Storer, "A Parallel Architecture for High Speed Data Compression", *Proceedings of the 3rd Symposium on the Frontiers of Massively Parallel Computation*, College Park, Maryland, October, 1990.

J. Reif and A. Yoshida, "Optical Expanders", manuscript, August 1989.

J. Reif, "Efficient Algorithms for Optical Computing with the DFT Primitive", *The 10th Conference on Foundations of Software Technology and Theoretical Computer Science*, Lecture Notes in Computer Science, Springer-Verlag, Bangalor, India, December 1990.

J. Reif, J. Canny, and A. Page, "An Exact Algorithm for Kinodynamic Planning in the Plane", *Symposium on Computational Geometry*, San Francisco, June, 1990.

J. Reif and S. Sen, "Random sampling techniques for binary search on fixed connection networks with applications to geometric algorithms", *ACM 2nd Annual Symposium on Parallel Algorithms and Architectures*, Crete, Greece, July,1990.

J. Reif, "Efficient Parallel Algorithms: Theory and Practice", *SIAM 35th Anniversary Meeting*, Denver, CO, Oct. 1987. Also *XI World Computer Congress*, IFIP 89, San Francisco, CA , 1989.

J. Reif and H. Djidjev, "An Efficient Algorithm for the Genus Problem", draft, April 1989.

J. Reif and H. Gazit, "A Parallel Planar Graph Isomorphism Algorithm", *ACM 2nd Annual Symposium on Parallel Algorithms and Architectures*, Crete, Greece, July, 1990.

J. Reif and V. Pan, "Acceleration of Minimum Cost Path Calculations in Graphs Having Small Separator Families", submitted for publication, 1990.

J. Reif and S. Sen, "Randomized Parallel Algorithms", workshop on Capabilities and Limitations of Parallel Computing, IBM, San Jose, CA, Dec. 1988. Also in *Information Processing* 89, G. Ritter (ed) Elsevier Science Publishers, North Hollond, IFIP, 1989, pp. 455-458 , and as Randomization in Parallel Algorithms and its Impact on Computational Geometry, in Optimal Algorithms, H. Djidjev editor, Springer-Verlag Lecture Notes in Computer Science 401, 1989, 1-8.

J. Reif, R. Paturi, and S. Rajasekaran, "The light bulb problem", presented at Workshop on Computational Learning Theory as Efficient and Robust Learning Using Statistical Bootstrap, Morgan Kaufmann Pub., Santa Cruz, CA, Aug. 1989. Submitted for journal publication.

J. Reif, D. Tygar, and A. Yoshida, "The Computation Complexity of Optical Beam Tracing", *31th IEEE Symposium on Foundations of Computer Science,*, Saint Louis, Missouri, October, 1990.

J. Reif and A. Tyagi, "Energy Complexity of Optical-Computations", to appear in, *The 2nd IEEE Symposium on Parallel and Distributed Processing*. December, 1990.

J. Reif, D.W. Blevins, E.W. Davis, and R.A. Hector, "BLITZEN: a highly integrated, massively parallel machine", *2nd Symposium on Frontiers of Massively Parallel Computation,* Fairfax, VA, Oct. 1988. Also in *Journal of Parallel and Distributed Computing*, Feb. 1990.

J. Reif and V. Ramachandran, "An optimal parallel algorithm for planarity (with V. Ramachandran). 30th Annual Symposium on Foundations of Computer Science, Durham, NC, Oct, 1989, pp. 282-287. Also University of Texas at Austin Technical Report TR-90-15, June 1990. Also invited to special issue of Journal of Algorithms,1990.

J. Reif and V. Pan, "Fast and efficient parallel solution of dense linear systems", *Computers and Mathematics with Applications* vol. 17, no. 11, pp. 1481-1491, 1989.

J. Reif and P. Gacs, "A simple three-dimensional real-time reliable cellular array", *17th Annual Symposium on Theory of Computing*, 388-395, Providence, RI, 1985.Accepted for publication by *Journal of Computer and System Sciences,* vol. 36, no. 2, p. 125-147, April 1990.

6. Individual Contributions

A. Individual Contribution (as Engineer/Scientist)

Reif particularly excels in investigating new emerging areas in computer science, which require the development of new theoretical models and efficient algorithms. He has succeeded in discovering highly innovative, new techniques to solve the key problems arising in these new areas. For example, Reif used innovative techniques to develop the most efficient known parallel algorithms for a large number of fundamental problems, including sorting, LR(K), parsing, graph problems such as planarity testing, solution of linear systems, etc.

Perhaps the most innovative and successful technique Reif has developed is the use of randomization (or coin flipping) to improve the performance of parallel algorithms. Examples of Reif's randomized parallel algorithms are (i) FLASHSORT, a parallel sorting algorithm which was theoretically optimal and has very efficient implementations on various massively parallel machines, and also (ii) the TREE CONTRACTION algorithm which has been used for efficient parallel solution of a large number of graph problems.

B. Significance of this Contribution

Reif has made major contributions to a wide range of other fundamental theoretical computer science problems in combinatorics, graph theory, algebra and game theory. He has focused particularly in emerging new areas such as parallel randomized algorithms.

This use of randomization had wide impact in the emerging field of parallel computing. Often the use of randomization simplified the algorithms, and thus made them much more practical to implement on actual parallel machines.

Reif's research productivity (52 journal papers and 81 conference papers) and national impact are substantially above the norm for a Senior IEEE member.

C. Further Contributions

As another example, Reif wrote an early paper 12 years ago on robotic movement, showing the first known PSPACE lower bounds on the problem, that lead to a subsequent large amount of theoretical work in robotics. He has done considerable further work in robotics, notably kinodynamic planning.

Secondary Contributions:
Although primarily a theoretical computer scientist, Reif also has made contributions to practical areas of computer science including parallel architectures, robotics, data compression and optical computing. In addition to having written theoretical papers in these areas, Reif also recently led a number of practical systems projects including:
(1) the implementation of parallel algorithms on massively parallel machines such as the CONNECTION and MPP machines,
(2) the architecture design of BLITZEN, a new massively parallel machine being constructed at the Microelectronics Center of North Carolina (MCNC);
(3) design and construction of very high rate parallel data compression hardware;
(4) invention of a very high rate electro-optic device for parallel message routing holographic addressing; and
(5) the development of a new Common Prototyping Language (Proteus) to be used for prototyping parallel and distributed algorithms.